

--- Auto generated Roll, Account Number, Message ID etc...

```
CREATE OR REPLACE TRIGGER TrigerForStudent
BEFORE INSERT ON STUDENT
FOR EACH ROW
```

```
DECLARE
i NUMBER;
BEGIN
SELECT NVL(COUNT(roll),0) INTO i FROM student;
```

```
i := i+1;
:NEW.ROLL := 'S' || LPAD(TO_CHAR(i),'6','0');
```

```
END;
/
```

```
INSERT INTO STUDENT(NAME,SEM,PAPER,YEAR,MARK,GARDE) VALUES('Ramesh','SEM II','Paper
5','2024',89,'A Grade');
```

```
select * from student;
desc student;
```

-- Find the odd / Even Balance Count --

```
DECLARE
sTemp EMP.SAL%TYPE;
eCount NUMBER;
oCount NUMBER;
CURSOR C IS select SAL from emp;
```

```
BEGIN
```

```
oCount := 0;
eCount := 0;
```

```
OPEN C;
LOOP
FETCH C INTO sTemp;
EXIT WHEN C%NOTFOUND;
```

```
IF (MOD(sTemp,2)=0) THEN
    eCount := eCount + 1;
ELSE
    oCount := oCount + 1;
```

```
END IF;
END LOOP;
CLOSE C;
DBMS_OUTPUT.PUT_LINE('Even Balance = ' || eCount);
```

```
DBMS_OUTPUT.PUT_LINE('Odd Balance = ' || oCount);  
END;  
/
```

```
select * from emp;
```

```
-- Declare the Package with name MyPck and it contain the function fact()
```

```
CREATE OR REPLACE PACKAGE MyPck AS
```

```
    FUNCTION fact(x number) RETURN number;
```

```
END MyPck;
```

```
-- Create the body of package and it's function body
```

```
CREATE OR REPLACE PACKAGE BODY MyPck AS
```

```
    FUNCTION fact(x number)
```

```
    RETURN number
```

```
    IS
```

```
        f number;
```

```
    BEGIN
```

```
        IF x=0 THEN
```

```
            f := 1;
```

```
        ELSE
```

```
            f := x * fact(x-1);
```

```
        END IF;
```

```
    RETURN f;
```

```
    END fact;
```

```
END MyPck;
```

```
-- Main Driver code
```

```
DECLARE
```

```
    num number;
```

```
    factorial number;
```

```
BEGIN
```

```
    num:= 5;
```

```
    factorial := MyPck.fact(num);
```

```
    dbms_output.put_line(' Factorial ' || num || ' is ' || factorial);
```

```
END;
```

```
/
```

```
Check This date is weak date or not
```

```

CREATE OR REPLACE PROCEDURE MYProcedure IS
    v_current_date DATE;
    v_day_number NUMBER;
    v_username VARCHAR2(30);
    vcnt NUMBER;
    ivcnt NUMBER;
BEGIN

    v_current_date := SYSDATE;

    DBMS_OUTPUT.PUT_LINE('Current Date: ' || TO_CHAR(v_current_date, 'DD-MON-YYYY
HH24:MI:SS'));

    v_day_number := TO_NUMBER(TO_CHAR(v_current_date, 'D'));

    IF v_day_number IN (1, 7) THEN
        DBMS_OUTPUT.PUT_LINE('It is the weekend. ');
    ELSE
        DBMS_OUTPUT.PUT_LINE('It is a weekday. ');
    END IF;

    v_username := USER;
    DBMS_OUTPUT.PUT_LINE('The Database User Name = ' || v_username);

    SELECT COUNT(*) INTO vcnt
    FROM ALL_OBJECTS
    WHERE OBJECT_TYPE = 'PROCEDURE' AND STATUS = 'VALID' ;

    SELECT COUNT(*) INTO ivcnt
    FROM ALL_OBJECTS
    WHERE OBJECT_TYPE = 'PROCEDURE' AND STATUS = 'INVALID';

    DBMS_OUTPUT.PUT_LINE('Valid Procedure Count: ' || vcnt);
    DBMS_OUTPUT.PUT_LINE('Invalid Procedure Count: ' || ivcnt);

END MYProcedure;
/

BEGIN
    MYProcedure;
END;
/

```

Triger for Auto Gen the Data of the table

```

CREATE OR REPLACE TRIGGER TrigerForOrder
AFTER INSERT ON ORDERS
FOR EACH ROW
DECLARE
    NUM NUMBER;
    newQ NUMBER;
BEGIN
    select qunt into NUM from product where pno = :NEW.PNO;
    IF (:NEW.QUNT < NUM or :NEW.QUNT = NUM) THEN
        newQ := NUM - :NEW.QUNT ;
    
```

```
        update product set qunt = newQ where pno = :NEW.PNO;  
ELSE  
    DBMS_OUTPUT.PUT_LINE('Requesting Item is not present');
```

```
END IF;  
END;  
/
```

```
insert into orders values(1,1,1,1);  
delete orders;  
select * from product;  
select * from orders;  
desc orders;  
select * from customer;
```

```
update product set qunt = 10 where pno = 1;
```